

Multi-Block Interleaved Codes for Local and Global Read Access

Yuval Cassuto*, Evyatar Hemo*, Sven Puchinger[†] and Martin Bossert[†]

*Viterbi Electrical Engineering Department, Technion – Israel Institute of Technology, Israel

[†]Institute of Communications Engineering, Ulm University, Germany

ycassuto@ee.technion.ac.il, evyatar.hemo@gmail.com, {sven.puchinger,martin.bossert}@uni-ulm.de

Abstract—We define multi-block interleaved codes as codes that allow reading information from either a small sub-block or from a larger full block. The former offers faster access, while the latter provides better reliability. We specify the correction capability of the sub-block code through its gap t from optimal minimum distance, and look to have full-block minimum distance that grows with the parameter t . We construct two families of such codes when the number of sub-blocks is 3. The codes match the distance properties of known integrated-interleaving codes, but with the added feature of mapping the same number of information symbols to each sub-block. As such, they are the first codes that provide read access in multiple size granularities and correction capabilities.

I. INTRODUCTION

The two central features sought in data-storage applications are *extreme data reliability* and *fast data access*. In data reliability we wish to avoid data loss in all conceivable circumstances, and in fast data access we want to read data with high throughput and low latency. Access considerations often dictate implementing an error-correcting code over a fixed (and not too large) data unit, which degrades the coding performance and the resulting reliability. An especially interesting instance of this happens in Flash storage, where a group of multi-level memory cells is divided to smaller data units called *pages*.

It is well known that coding over large block lengths gives the best reliability for a given coding rate. But adding access performance to the considerations, block lengths are tightly constrained by the granularity required for the read/write interface of the storage device. The standard approach of the storage industry is to fix the coding block length to be the basic access unit of the device (e.g. a page of a few KB), and come up with the best code for this block length. A conflict between coding efficiency and access performance thus emerges whenever the storage is required to deliver data at fine access granularities that imply short code blocks. Solving this conflict needs a multi-level read/write scheme with the following features:

- 1) Jointly writing m data (sub-)units (e.g. pages), each with k symbols, to a single write block.
- 2) Allowing random read access to a sub-unit, i.e., returning its k symbols by physically reading a sub-block $1/m$ smaller than the write block.
- 3) In cases where a sub-unit read fails due to excess errors in the sub-block, reading the full write block succeeds in retrieving the data.

Feature 2 takes care of the fast-access requirement, and feature 3 improves reliability with only a minor adjustment of read performance in rare unfortunate instances. Note that focusing on random sub-unit *read* performance is in line with real applications that are much more sensitive to delayed reads than writes.

The formal model that fits the above three features now follows. Define a *write block* as $N = mn$ symbols, where m is the number of *sub-blocks* and n is the number of symbols in each sub-block. A *write unit* of $K = mk$ information symbols divides into m *sub-units* of size k each. In the write path, K information symbols are encoded into N code symbols and written to the memory. In the read path, we need to return a sub-unit of k symbols by reading a sub-block of n symbols. This operation is divided into *decoding* and *reverse mapping*, where the former corrects the errors/erasures in the n sub-block symbols, and the latter retrieves the k information symbols from the corrected sub-block. If sub-block decoding failed to correct the errors/erasures, we allow reading the entire write block and decode it as $[N, K]$ code. We define an $\langle N, K, m \rangle$ *multi-block interleaved code* as a code supporting these operations with the parameters N, K, m . Our objective in this paper is to construct $\langle N, K, m \rangle$ codes with good distance properties for both individual sub-block decoding and full write-block decoding.

The first place to look for multi-block interleaved codes is within the literature on *integrated-interleaving (I-I) codes* [6], [5] (see also [1]), which provide one minimum distance for sub-block codewords and a larger one for full codewords. While I-I codes give good (sometimes provably optimal) tradeoffs between sub-block and full-codeword distances, they cannot be readily used as multi-block interleaved codes. The issue with I-I codes is that it is not known how to reverse-map a sub-block codeword to k information symbols. The encoder presented for I-I codes [1] maps different numbers of information symbols to different sub-blocks, and it is not clear how to re-organize the resulting $mk \times mn$ generator matrix to get m full-rank $k \times n$ sub-matrices. Because of that, I-I codes fail to support the critical feature 2 in the read/write model. Another related coding framework is *locally recoverable codes (LRC)* [4], [7], [10]. LRC are focused on efficient local *repair* of individual code symbols, while here we are interested in local decoding of larger sub-blocks. The primary reason why known LRCs are not directly applicable to the proposed model is that they partition the code coordinates to disjoint sets that are each an MDS local code. Making each of the m sub-blocks

an $[n, k]$ MDS local code degenerates the problem, because all parity symbols are local in this case.

Our main contributions in the paper are two code constructions of multi-block interleaved codes providing flexible tradeoffs between sub-block and full-codeword correction capabilities. The flexibility is achieved by introducing an integer parameter t specifying the gap of the sub-block code from minimum-distance optimality. As t grows, the constructed codes enjoy better minimum distances for the full write-block codewords. This new view of the local vs. global correction tradeoff through the t parameter allows their explicit joint-optimization, while in I-I codes this relation is much less transparent (in I-I codes there is no notion of sub-block dimension k , so t is not even defined). One family of codes, presented in Section III, allows improvement in the full-block minimum distance (by growing t) reaching $d = 1.5(n - k + 1)$. An improved construction, presented in Section IV, can grow the full-block minimum distance further until $d = 1.6(n - k + 1)$. The proposed constructions are given for $m = 3$ sub-blocks, but can be extended to additional m with similar techniques. The key idea toward having sub-block reverse mapping is the specification of the codes through their generator matrices. This enables to carefully populate the generator matrix with constituent Reed-Solomon (RS) generators such that all distance properties are fulfilled simultaneously. Coincidentally, both constructions achieve the same distance properties as known I-I codes: the former meets 2-level I-I and the latter meets 3-level I-I. However, the constructed codes are *not* reformulations of these known I-I codes: the resulting codes are different, and there is no apparent transformation from the I-I codes to our codes that adds the desired sub-block access features. A previous use of constituent RS generators was in [3] to construct *partial unit memory (PUM)* convolutional codes [9], [8] (see also [2]).

II. DEFINITIONS AND NOTATIONS

Throughout the paper we denote the set $\{a, a + 1, \dots, b\}$ by $[a : b]$. Similarly, $[a : b]_n$ denotes the same set, but with element indices taken modulo n . For example, if $n = 7$, then $[5 : 9]_n = [5 : 2]_n = \{5, 6, 0, 1, 2\}$. The operation $A \setminus B$ represents set difference between A and B . We use the term *distance* to refer to the Hamming distance between vectors over the field $\text{GF}(q)$. For notational convenience, the statements in the paper on code minimum distances are written as some d equal some number, even though in some cases we only prove (the important part) that d is *at least* that number. We first define our principal object of study, which we call *multi-block interleaved code*.

Definition 1.[multi-block interleaved code] An $[N, K]$ linear code C is a $\langle N, K, m \rangle$ **multi-block interleaved code** if its code coordinates are partitioned to m disjoint sub-blocks of size $n = N/m$ each, and from each sub-block we can recover a disjoint size- $k = K/m$ sub-unit of information symbols.

For decoding a multi-block interleaved code we assume m projections of the coordinates $[1 : N]$ to disjoint subsets $[1 : n]$, $[n + 1 : 2n]$, ... up to $[(m - 1)n + 1 : mn]$. We assume that the projection of C onto any of these subsets gives the same $[n, \ell]$

code, which we denote C' . Next we define the distances of multi-block interleaved codes.

Definition 2.[sub-block minimum distance] An $\langle N, K, m \rangle$ multi-block code C has **sub-block minimum distance** δ if the sub-block projected code C' has minimum distance δ .

The (full-block) minimum distance of an $\langle N, K, m \rangle$ multi-block code will be defined to be the usual minimum distance as $[N, K]$ block code.

A. Bound on the minimum distance of multi-block interleaved codes

The following is a restatement of a known theorem from [5], but posed in different notation (using the t parameter) that is helpful for the constructions in the sequel.

Theorem 1. Let C be an $\langle N = mn, K = mk, m \rangle$ multi-block interleaved code with sub-block minimum distance $\delta = n - k - t + 1$. If $t \leq (k - 1)/(m - 1)$ then the minimum distance of C is bounded by

$$d \leq n - k + (m - 1)t + 1. \quad (1)$$

An optimal multi-block interleaved code is one meeting the bound (1) with equality. We note two interesting special cases of Theorem 1. For $m = 1$ we get the usual Singleton bound; for $m = 2$ we get $d \leq n - k + t + 1$ when $t \leq k - 1$. The upper bound reveals the tradeoff between the sub-block minimum distance and the minimum distance of the code. The parameter t prescribes the gap of the sub-block distance from optimality, and in return increasing t improves the upper bound on the distance of the full-block code.

B. Short review of Reed-Solomon codes

The key building block for our constructions is Reed-Solomon (RS) codes, which are now defined.

Definition 3.[Reed-Solomon code] Let α be an element of a finite field $\text{GF}(q)$ of order n . Then define a $[n, \ell]$ **Reed-Solomon code** as all polynomials $c(x)$ obtained as

$$c(x) = i(x)g(x),$$

where $i(x)$ is an arbitrary information polynomial of degree $\leq \ell - 1$, and

$$g(x) = (x - \alpha^{-s})(x - \alpha^{-(s+1)}) \cdots (x - \alpha^{-(s+n-\ell-1)}), \quad (2)$$

and s is some integer in $[0 : n - 1]$.

In simple words, an RS code is obtained by a generator polynomial $g(x)$ with $n - \ell$ roots whose reciprocals are consecutive powers of α . As a useful compact notation, given n and q , we define a RS code through its (cyclically) consecutive power set. In this representation, the code defined by the generator in (2) is

$$RS([s : s + r - 1]_n),$$

where $r = n - \ell$ is called the *redundancy* of the code. It is well known that the minimum distance of $RS([s : s + r - 1]_n)$ is $r + 1$. It will often be convenient to represent an RS code by the *complement* of its root power set, that is,

$$\overline{RS}([a : b]_n) \triangleq RS([0 : n - 1]_n \setminus [a : b]_n).$$

The *dimension* of the code $\overline{RS}([a : a + \ell - 1]_n)$ is ℓ , and its minimum distance is $n - \ell + 1$.

III. CONSTRUCTION OF MULTI-BLOCK INTERLEAVED CODES

Our focus in the paper is on codes with $m = 3$, because this is the most useful case for Flash-based storage devices with $8 = 2^3$ representation levels. The constructions can be generalized to larger m values. For the specific case of $m = 3$ we add the following definitions.

Definition 4. [D_i -minimum-distances] For $i \in \{1, 2, 3\}$, an $\langle N, K, 3 \rangle$ multi-block interleaved code C has D_i -**minimum-distance** d_i if the lowest weight of any codeword with exactly i non-zero sub-blocks is d_i .

Note that according to Definition 4, the minimum distance of a $\langle N, K, 3 \rangle$ code is $\min_{i \in \{1, 2, 3\}} d_i$. In addition to serving as upper bounds on the code minimum distance, the D_i -minimum-distances have important operational meanings: the D_1 -minimum-distance gives the correction capability when 1 sub-block suffers many errors/erasures, and the D_2 -minimum-distance does the same when 2 sub-blocks suffer many errors/erasures.

Construction 1. Let C_3 be defined by a generator matrix $G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}$, where G_1, G_2 and G_3 are the $k \times 3n$ matrices given in

Fig. 1. Now we define the component matrices of Fig. 1; blank rectangles represent all-zero sub-matrices.

- G_1 is a generator matrix for the code $\overline{RS}([0 : t-1]_n)$.
- G_2 is a generator matrix for the code $\overline{RS}([t : 2t-1]_n)$.
- G_1 is a generator matrix for the code $\overline{RS}([2t : k-1]_n)$.
- GE is a generator matrix for the code $\overline{RS}([k : k+t-1]_n)$.

		n	n	n
$G_1 =$	$k-2t$	GI		
	t	$G1$		GE
	t	$G2$	GE	
$G_2 =$	$k-2t$		GI	
	t	GE	$G1$	
	t		$G2$	GE
$G_3 =$	$k-2t$			GI
	t		GE	$G1$
	t	GE		$G2$

Figure 1. Generator matrices for a $\langle 3n, 3k, 3 \rangle$ multi-block interleaved code C_3 .

First note that the dimensions of the codes specified for $G1, G2, GI, GE$ fit their corresponding sub-matrices in Fig. 1. It is straightforward to write a generator matrix for any code of the type $\overline{RS}([a : a + \ell - 1]_n)$, because the codewords of this code are evaluations of polynomials of the form $x^a U(x)$, where $U(x)$ is a polynomial of degree $\ell - 1$ or less. As required,

the dimension of C_3 is $3k$, and its length is $3n$. We defer the discussion on encoding and decoding till after we prove the minimum distances in the following.

Proposition 2. When $t < k/2$, C_3 has sub-block minimum distance $\delta = n - k - t + 1$.

Proof: Looking at any length- n sub-block, we observe from the structure of G that the codeword projected on the sub-block is a linear combination of codewords generated by $G1, G2, GI, GE$. Noting the root power sets of these codes (the complements of the sets listed in Construction 1), we get: $G1 \rightarrow \overline{RS}([t : n-1]_n)$, $G2 \rightarrow \overline{RS}([2t : t-1]_n)$, $GI \rightarrow \overline{RS}([k : 2t-1]_n)$, and $GE \rightarrow \overline{RS}([k+t : k-1]_n)$. If a power index lies in the intersection of these sets, then every polynomial of a codeword in the sub-block code has this power index as root. Taking the intersection of these sets we obtain that the projected sub-block code is a

$$\overline{RS}([k+t : n-1]_n)$$

code, and thus its minimum distance is $n - k - t + 1$. ■

Given the sub-block minimum distance of Proposition 2, the next lemma shows that the D_1 -minimum-distance of Construction 1 is optimal with respect to the bound¹ of Theorem 1.

Lemma 3. A code C_3 from Construction 1 has D_1 -minimum-distance $d_1 = n - k + 2t + 1$, for all $t < k/2$.

Proof: Denote the length $3k$ information vector as \mathbf{v} , and partition it as

$$\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3).$$

Each sub-vector \mathbf{v}_j has length k , and we further partition it to

$$\mathbf{v}_j = (\mathbf{v}_{j,1}, \mathbf{v}_{j,2}, \mathbf{v}_{j,3}).$$

The lengths of the $m = 3$ constituent vectors are from left to right: $k - 2t, t, t$. Consider a codeword of C_3 with one non-zero sub-block. Assume (wlog due to symmetry) that the left sub-block is non-zero. It is clear from the structure of G in Fig. 1 that all sub-vectors except $\mathbf{v}_{1,1}$ must be all-zero. In this case, the codeword weight is at least the minimum weight of a non-zero codeword from $\overline{RS}([0 : k-1]_n)$ (spanned by GI), which is $n - k + 2t + 1$ as required by the lemma statement. ■ The behavior of the D_2 -minimum-distance as a function of t is quite different, as we now see.

Lemma 4. A code C_3 from Construction 1 has D_2 -minimum-distance $d_2 = 2(n - k - t + 1)$.

The proof of Lemma 4 is trivial from the fact that $\delta = n - k - t + 1$. Unfortunately, the construction does not exclude codewords with two minimum-weight sub-block codewords. The problem lies in the fact that both $\mathbf{v}_{1,1}$ and $\mathbf{v}_{2,2}$ are multiplied by the same matrix GE in the right column, which may cancel their contribution to the right sub-block. We now summarize the distance properties in the following theorem (proof immediate).

¹The bound is given on the minimum distance, but from the proof of Theorem 1 it readily applies to the D_1 -minimum-distance as well.

Theorem 5. A code C_3 from Construction 1 has sub-block minimum distance $\delta = n - k - t + 1$, D_1 -minimum-distance $d_1 = n - k + 2t + 1$, and D_2 -minimum-distance $d_2 = 2(n - k - t + 1)$. For $t \leq (n - k + 1)/4$, C_3 has minimum distance $d = d_1$ (optimal), and for $(n - k + 1)/4 < t$ it has minimum distance $d = d_2$ (not optimal).

Similar to the case here, all the known I-I constructions in the literature with optimal D_1 -minimum-distance (given t) suffer from the same problem of having codewords with two minimum-weight sub-block codewords. In [5] the authors address this exact problem, but give no explicit constructions. The implication of Theorem 5, in particular the terms $2t$ in d_1 and $-2t$ in d_2 , is that there is a ratio of 1 between the loss in d_2 to the gain in d_1 as we increase t . This high ratio compromises the code's ability to address high error/erasure events in both one and two sub-blocks. In our next construction we solve this by proving a much lower ratio of $2/3$. But before that, we want to show the good property of Construction 1 having an extremely efficient reverse mapping from the sub-block codeword to the k information symbols of the input sub-unit \mathbf{v}_j . We now write the encoding and reverse-mapping functions explicitly.

Encoding: The encoder input is an information vector $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, where each \mathbf{v}_j is a vector of k $\text{GF}(q)$ symbols holding the information sub-unit for sub-block j . The output of the encoder is simply the vector $\mathbf{v} \cdot G$, which is a codeword of C_3 . The matrices $G_1, G_2, G_3, G_4, G_1, G_2, G_3, G_4, G_1, G_2, G_3, G_4$ are chosen as the standard polynomial-evaluation matrices of the respective RS codes (mapping ℓ polynomial coefficients to n code symbols). For a code $\overline{RS}([a : a + \ell - 1]_n)$, the corresponding generator matrix maps an information sub-vector \mathbf{u} to a codeword by evaluating the polynomial $\tilde{U}(x) = x^a U(x)$ on n elements of $\text{GF}(q)$, where $U(x)$ is the polynomial whose coefficients are the elements of \mathbf{u} , and it has degree at most $\ell - 1$. Note that this is not the way the RS code was defined in Definition 3, but it is true by an equivalent definition of RS codes.

Reverse mapping: The reverse-mapping operation returns the information sub-unit \mathbf{v}_j from the codeword of sub-block j . Denote by $u(x)$ the polynomial whose coefficients are the codeword symbols corresponding to \mathbf{u} . From the property given in Definition 3, $u(\alpha^{-i})$ is zero for all i except those in $[a : a + \ell - 1]_n$. Since these sets are not overlapping between $G_1, G_2, G_3, G_4, G_1, G_2, G_3, G_4$, we can take the combined sub-block codeword polynomial $c(x)$ (which is a sum of different polynomials like $u(x)$ above), and get that for each $i \in [0 : k - 1]$, that $c(\alpha^{-i}) = u(\alpha^{-i})$ for one of the sub-vectors \mathbf{u} in \mathbf{v}_j . Now we can use the well known property of RS codes (inverse DFT) and obtain

$$U_l = n^{-1} u(\alpha^{-(a+l)}) = n^{-1} c(\alpha^{-(a+l)}), \quad l \in [0 : \ell - 1].$$

Since $c(x)$ is given, we can find the coefficients of all the polynomials $U(x)$ and thus all the vectors \mathbf{u} .

Neither the I-I codes from [6] nor the ones in [5] have a known way to map k information symbols to each sub-block codeword, such that it is possible to retrieve these k symbols back from the sub-block codeword. The encoder specified for I-I codes [1] (corresponding to the parameters of

Construction 1) maps $k + t$ information symbols to each of two sub-blocks, and $k - 2t$ to the third.

IV. AN IMPROVED CONSTRUCTION WITH LARGER MINIMUM DISTANCE

To get codes $\langle N, K, m \rangle$ with larger minimum distances beyond what Construction 1 can achieve, we present the following improved construction. In the sequel we define $s = t/2$, assuming even t .

Construction 2. Let \mathcal{K}_3 be defined by a generator matrix $G = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}$, where G_1, G_2 and G_3 are $k \times 3n$ matrices specified in Fig. 2 as follows. The figure specifies $4s$ rows of each G_i , which are populated by the generator matrices specified below, or combinations thereof. The figure omits the top $k - 4s$ rows of each G_i that host the GI matrix given below, in the corresponding sub-block².

- G_1 is a generator matrix for the code $\overline{RS}([0 : s - 1]_n)$.
- G_2 is a generator matrix for the code $\overline{RS}([s : 2s - 1]_n)$.
- G_3 is a generator matrix for the code $\overline{RS}([2s : 3s - 1]_n)$.
- G_4 is a generator matrix for the code $\overline{RS}([3s : 4s - 1]_n)$.
- GI is a generator matrix for the code $\overline{RS}([4s : k - 1]_n)$.
- GE is a generator matrix for the code $\overline{RS}([k : k + s - 1]_n)$.
- GF is a generator matrix for the code $\overline{RS}([k + s : k + 2s - 1]_n)$.

	n	n	n	
$\mathbf{v}_{1,1} :$	G_1	GF		$4s$
$\mathbf{v}_{1,2} :$	G_2	GE		
$\mathbf{v}_{1,3} :$	G_3		GE	
$\mathbf{v}_{1,4} :$	$G_4 + G_3$		GE	
$\mathbf{v}_{2,1} :$		G_1	GF	$4s$
$\mathbf{v}_{2,2} :$		G_2	GE	
$\mathbf{v}_{2,3} :$	GE	G_3		
$\mathbf{v}_{2,4} :$	GE	$G_4 + G_3$		
$\mathbf{v}_{3,1} :$	GF		G_1	$4s$
$\mathbf{v}_{3,2} :$	GE		G_2	
$\mathbf{v}_{3,3} :$		GE	G_3	
$\mathbf{v}_{3,4} :$		GE	$G_4 + G_3$	
	left	center	right	

Figure 2. Generator matrices for an improved $\langle 3n, 3k, 3 \rangle$ multi-block interleaved code \mathcal{K}_3 .

As in the previous construction the dimensions of the codes specified for $G_1, G_2, G_3, G_4, G_1, G_2, G_3, G_4, G_1, G_2, G_3, G_4$ fit their corresponding sub-matrices in Fig. 2. The code dimension of \mathcal{K}_3 is $3k$, and its length is $3n$. We prove the minimum distances in the following.

Proposition 6. When $t < k/2$, \mathcal{K}_3 has sub-block minimum distance $\delta = n - k - t + 1$.

Proof: The proof is identical to Proposition 2: at each sub-block there is a codeword of an RS code with $n - k - 2s = n - k - t$ roots with consecutive powers. ■

²same as in Fig. 1.

Lemma 7. *The code \mathcal{K}_3 has D_1 -minimum-distance $d_1 = n - k + 3t/2 + 1$.*

Proof: Recall the partition $\mathbf{v} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$. Now we partition each \mathbf{v}_j to

$$\mathbf{v}_j = (\mathbf{v}_{j,1}, \mathbf{v}_{j,2}, \mathbf{v}_{j,3}, \mathbf{v}_{j,4}).$$

The lengths of the constituent sub-vectors are from left to right: $k - 4s, s, s, s, s$. Consider a codeword of \mathcal{K}_3 with one non-zero sub-block, for which we seek a lower bound on the Hamming weight. Assume (wlog due to symmetry) that the one non-zero sub-block is the left one. Then it is clear from Fig. 2 that the only sub-vectors that can be non-zero are $\mathbf{v}_{1,1}$, $\mathbf{v}_{1,3}$, and $\mathbf{v}_{1,4}$. Setting any other sub-vector to non-zero implies a non-zero codeword in at least one other sub-block. Further, if $\mathbf{v}_{1,3}$ or $\mathbf{v}_{1,4}$ are non-zero, then they must satisfy $\mathbf{v}_{1,3} + \mathbf{v}_{1,4} = 0$. Otherwise the codeword in the right sub-block will be non-zero as well. It can be seen that this last condition implies that only $G4$ contributes to the non-zero codeword, and together with the contribution of $G1$ from $\mathbf{v}_{1,1}$, we get $n - k + 3s$ consecutive roots and $d_1 = n - k + 3t/2 + 1$. ■

Lemma 8. *The code \mathcal{K}_3 has D_2 -minimum-distance $d_2 = 2(n - k - t/2 + 1)$.*

Proof: We use the definition of \mathbf{v} from the proof of Lemma 7. Consider a codeword of \mathcal{K}_3 with two non-zero sub-blocks, for which we seek a lower bound on the total Hamming weight. Assume (wlog due to symmetry) that the two non-zero sub-blocks are the left and center ones. Then it is clear from Fig. 2 that the only sub-vectors that can be non-zero are $\mathbf{v}_{1,1}$, $\mathbf{v}_{1,3}$, $\mathbf{v}_{1,4}$ from \mathbf{v}_1 , and $\mathbf{v}_{2,1}$, $\mathbf{v}_{2,2}$, $\mathbf{v}_{2,3}$, $\mathbf{v}_{2,4}$ from \mathbf{v}_2 . Other non-zero sub-vectors would imply a non-zero codeword on the right sub-block. The above restriction on the sub-vectors that can be non-zero implies that there is no contribution of F in the left sub-block codeword and no contribution of $G1$ in the center sub-block codeword. So each of the sub-block codewords has $n - k - s$ consecutive-power roots, and thus $d_2 = 2(n - k - s + 1)$. ■

Combining Proposition 6 and Lemmas 7,8, we summarize the distance properties of Construction 2 in the following theorem (proof omitted).

Theorem 9. *A code \mathcal{K}_3 from Construction 2 has sub-block minimum distance $\delta = n - k - t + 1$, D_1 -minimum-distance $d_1 = n - k + 3t/2 + 1$, and D_2 -minimum-distance $d_2 = 2(n - k - t/2 + 1)$. For $t \leq 2(n - k + 1)/5$, \mathcal{K}_3 has minimum distance $d = d_1$, and for $2(n - k + 1)/5 < t \leq (n - k + 1)/2$ it has minimum distance $d = d_2$.*

Theorem 9 demonstrates the value of Construction 2: it allows to use large values of t that give minimum distances higher than Construction 1 can reach. It can be calculated that when $t > 2(n - k + 1)/7$, Construction 2 has superior minimum distance compared to Construction 1. From $t = 2(n - k + 1)/7$ we can increase t in Construction 2 up to $t = 2(n - k + 1)/5$, and get minimum distance as high as $d = 1.6(n - k + 1)$, while Construction 1 has a cutoff at $d = 1.5(n - k + 1)$.

Erasure decoding: We chose to prove the code correction

properties by lower bounding the minimum distances. However, an alternative way is by giving an explicit erasure-decoding algorithm, which we do now. First, suppose that one sub-block has $n - k + 3t/2$ erasures and the other two have each $n - k - t$ erasures or less. From symmetry assume that the left sub-block has the largest number. Then decoding the center sub-block with a $[n, k + t]$ code gives $\mathbf{v}_{2,1}, \dots, \mathbf{v}_{2,4}$, and similarly the right block gives $\mathbf{v}_{3,1}, \dots, \mathbf{v}_{3,4}$. In addition, we get $\mathbf{v}_{1,1}$ as the reverse map of GF in the center sub-block, and then $\mathbf{v}_{1,2}$ by canceling $\mathbf{v}_{3,3}$ and $\mathbf{v}_{3,4}$ from the reverse map of GE in the center sub-block. Finally, we can find $\mathbf{v}_{1,3} + \mathbf{v}_{1,4}$ by canceling $\mathbf{v}_{2,2}$ from the reverse map of GE in the right sub-block. Now we can decode the left sub-block with a $[n, k - 3t/2]$ code having contributions from only $G1$ and $G4$. Second, suppose that two sub-blocks have each $n - k - t/2$ erasures and the third one has $n - k - t$ erasures or less. From symmetry assume that the right sub-block has the smaller number. Then decoding the right sub-block with a $[n, k + t]$ code gives $\mathbf{v}_{2,1}$, and $\mathbf{v}_{3,1}$ (among all $\mathbf{v}_{3,i}$). Now after canceling these two we can decode each of the left and center sub-blocks with a $[n, k + t/2]$ code and recover their erased symbols.

V. CONCLUSION

The constructions in this paper are the first known ones to offer read access in both sub-blocks and full blocks. The first construction has optimal D_1 -minimum-distance given t , but a limiting reduction in the D_2 -minimum-distance as t grows. The second construction has slower growth of the D_1 -minimum-distance, but in return also slower reduction in D_2 , and altogether a higher maximal minimum distance. Future work is needed to extend these constructions, and find others with similar tradeoffs, to additional m values. While the codes suffice to use finite fields with sizes as small as the sub-block size, further reducing the field size is an important future direction, e.g. through using BCH codes as the constituents.

VI. ACKNOWLEDGEMENT

This work was supported in part by the German-Israel Foundation and by the Israel Science Foundation.

REFERENCES

- [1] M. Blaum and S. Hertzler, "Integrated interleaved codes as locally recoverable codes: properties and performance," *Int. J. Inf. Coding Theory*, vol. 3, no. 4, pp. 324–344, January 2016.
- [2] M. Bossert, *Channel Coding for Telecommunications*. Chichester, UK: Wiley, 1999.
- [3] U. Dettmar and U. Sorger, "New optimal partial unit memory codes based on extended BCH codes," *Electronics letters*, vol. 29, no. 23, pp. 2024–2025, November 1993.
- [4] P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the locality of codeword symbols," *IEEE Trans. Inf. Theory*, vol. 58, no. 11, pp. 6925–6934, November 2012.
- [5] J. Han and L. Lastras-Montano, "Reliable memories with subline accesses," in *IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007.
- [6] M. Hassner, K. Abdel-Ghaffar, A. Patel, R. Koetter, and B. Trager, "Integrated interleaving – a novel ECC architecture," *IEEE Trans. Magn.*, vol. 37, no. 3, pp. 773–775, February 2001.
- [7] G. Kamath, N. Prakash, V. Lalitha, and P. V. Kumar, "Codes with local regeneration and erasure correction," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4637–4660, August 2014.
- [8] G. Lauer, "Some optimal partial-unit-memory codes (corresp.)," *IEEE Trans. Inf. Theory*, vol. 25, no. 2, pp. 240–243, March 1979.

- [9] L.-n. Lee, "Short unit-memory byte-oriented binary convolutional codes having maximal free distance (corresp.)," *IEEE Trans. Inf. Theory*, vol. 22, no. 3, pp. 349–352, May 1976.
- [10] I. Tamo and A. Barg, "A family of optimal locally recoverable codes," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4661–4676, August 2014.